UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/539,624	03/31/2000	Jerrie L. Coffman	219.38025X00	9576

7590 08/13/2004

Rob D. Anderson
C/O Blakely, Sokoloff, Taylor, & Zafman LLP
12400 Wilson Boulevard
Seventh Floor
Los Angeles, CA 90025

EXAMINER	
ALI, SYED J	

ART UNIT	PAPER NUMBER
2127	

DATE MAILED: 08/13/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 09/539,624	**Applicant(s)** COFFMAN ET AL.	
	Examiner Syed J Ali	**Art Unit** 2127	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *13 May 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-21* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-21* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

DETAILED ACTION

1. This office action is in response to the amendment file May 13, 2004. Claims 1-21 are presented for examination.

2. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

Claim Rejections - 35 USC § 102

3. **Claim 1 is rejected under 35 U.S.C. 102(b) as being anticipated by Lindholm et al. (USPN 5,797,004) (hereinafter Lindholm).**

4. As per claim 1, Lindholm teaches the invention as claimed, including a system comprising:

a shared resource (col. 1 lines 30-45);

multiple processors arranged to access said shared resource (col. 1 lines 16-29); and

an operating system configured to allow said multiple processors to perform work on said shared resource concurrently while supporting state changes or updates of said shared resources (col. 3 line 54 - col. 4 line 2; col. 4 lines 34-45), said operating system comprising a synchronization algorithm for synchronizing multiple threads of operation with a single thread so as to achieve mutual exclusion between multiple threads performing work on said shared resource (col. 4 lines 37-45; col. 6 lines 10-21) and a single thread updating or changing the state of said shared resource without requiring serialization of all threads (col. 5 lines 34-45; col. 6 lines 10-21) such that an update or change of the state of the shared resource may be made by the single thread only when none of the multiple threads are processing work on the shared resource (col. 6 lines 10-21; col. 7 lines 14-39; col. 11 lines 37-47).

Claim Rejections - 35 USC § 103

5. **Claims 2-3 and 9-10 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lindholm in view of Spix et al. (USPN 5,179,702) (hereinafter Spix).**

6. As per claim 2, Spix teaches the invention as claimed, including the following limitations not shown by Lindholm:

the system as claimed in claim 1, wherein said shared resource includes work queues associated with a hardware adapter configured to send and receive message data to/from a remote system (col. 15 lines 3-27).

7. It would have been obvious to one of ordinary skill in the art to combine Lindholm with Spix since the method disclosed in Spix, while disclosing a way of allowing multiple CPUs to access shared resources that include work queues, does so according to an 'anarchistic' scheduling algorithm, which may be unsuitable for most needs. Thus, the combination with Lindholm provides a rigid scheduling algorithm that regulates access to synchronization constructs, while utilizing shared resources of particular I/O devices that support work queues, such as those associated with network communication.

8. As per claim 3, Lindholm teaches the invention as claimed, including the system as claimed in claim 2, wherein said synchronization algorithm is executed to synchronize any thread wishing to update or change a state of said shared resource with all the threads processing I/O operations on said shared resource (col. 6 line 56 - col. 7 line 6).

9. As per claim 9, Lindholm teaches the invention as claimed, including the system as claimed in claim 2, wherein said synchronization algorithm is installed as part of a software driver module of an operation system [OS] kernel or an user-level application of said system (col. 3 lines 31-33).

10. As per claim 10, Spix teaches the invention as claimed, including the system as claimed in claim 2, wherein said shared resource includes one of work queues, completion queues, FIFO queues, hardware adapters, I/O controllers and other memory elements of said system (col. 15 lines 3-27).

11. Concerning the additional limitations of hardware adapters, I/O controllers, etc. Lindholm teaches that these types of shared resources that may require synchronization (col. 1 lines 30-45).

12. **Claims 4-7 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lindholm in view of Kerrigan et al. (USPN 5,404,488) (hereinafter Kerrigan).**

13. As per claim 4, Kerrigan teaches the invention as claimed, including the following limitations not shown by Lindholm:

the system as claimed in claim 1, wherein said synchronization algorithm is executed to allow worker threads to work concurrently while processing I/O operations in exclusion of an update thread when a state of said shared resource is not changing, and allow an update thread to change the state or update said shared resource in exclusion of multiple worker threads (col. 24 lines 28-39).

14. It would have been obvious to one of ordinary skill in the art to combine Lindholm with Kerrigan since Lindholm allows all threads that are synchronized with an object to perform state updates on the shared resource. This can lead to excessive context switching and tremendous overhead, a situation that requires remedy. Kerrigan teaches the invention as claimed, including a way of updating an application using an update thread and a synchronization construct, specifically a semaphore. Although Kerrigan is related to updating the data pertaining to an application, the idea is easily combinable with Lindholm since Kerrigan discloses threads and synchronization constructs, as does Lindholm. The combination therein would thus allow Lindholm to be modified such that only one specific thread performs updates on the shared resource, thereby reducing the costly overhead incurred if each thread working on a shared resource updated the status upon locking or unlocking the resource.

15. As per claim 5, Lindholm teaches the invention as claimed, including the system as claimed in claim 4, wherein said synchronization algorithm is executed to support a worker thread operation for processing simultaneous I/O operations on said shared

resource while concurrently supporting an update thread operation for updating or changing the state of said shared resource (col. 1 lines 30-45).

16. As per claim 6, Lindholm teaches the invention as claimed, including a system as claimed in claim 5, wherein said worked thread operation is invoked by one of an event and a user's request, and is performed by:

determining whether a lock is available (col. 5 lines 7-20);

if the lock is not available, waiting until the lock becomes available (col. 6 lines 22-34);

if the lock is available, seizing the lock while incrementing a count by a discrete constant to indicate the number of worker threads that are active, and then releasing the lock after the count has been incremented (col. 11 line 65 - col. 12 line 3);

after the lock has been released, allowing multiple worker threads to process work concurrently (col. 3 line 55 - col. 7 line 20);

determining next whether there is work to be processed (col. 7 lines 23-29);

if there is work to be processed, processing the work until there is no work to be processed (col. 7 lines 23-29; and

if there is no work to be processed, decrementing the count by a discrete constant to indicate when all the worker threads are done with completion processing (col. 7 line 47 - col. 8 line 19).

17. As per claim 7, Lindholm teaches the invention as claimed, including a system as claimed in claim 6, wherein said update thread operation is invoked by a user's request, and is performed by:

determining whether a lock is available (col. 5 lines 7-20);

if the lock is not available, waiting until the lock becomes available when released by any one of the worker threads (col. 6 lines 22-34);

if the lock is available, seizing the lock until the count becomes zero (0) to indicate that it is safe to update or change the state of said shared resource, and updating or changing the state of said shared resource (col. 7 line 63 - col. 8 line 19); and

after said shared resource has been updated, releasing the lock so as to allow either new worker threads to continue I/O operation processing or a different update thread to continue shared resource updating (col. 7 line 63 - col. 8 line 19).

18. **Claims 8, 11-12, and 17-19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lindholm in view of Spix in view of Tillier (previously cited).**

19. As per claim 8, Tillier teaches the invention as claimed, including the following limitations not shown by the modified Lindholm, specifically a system as claimed in claim 2, further comprising data channels formed between said system and said remote system, via a switched fabric, and supported by the "*Virtual Interface [VI] Architecture Specification*" and the "*Next Generation Input/Output [NGIO] Specification*" for message data transfers between said system and said remote system (col. 5 lines 14-36).

20. It would have been obvious to one of ordinary skill in the art to combine Lindholm and Spix with Tillier since both Lindholm and Spix only speak to synchronization of threads in a multiprocessor system, but does not necessarily account for other types of networks, such as a switched fabric network. Tillier teaches a way of implementing such systems and thus would allow the modified Lindholm to be implemented on a wider variety of systems.

21. As per claim 11, Lindholm teaches the invention as claimed, including an operating system configured to allow said multiple processors to perform work on said shared resource concurrently while supporting state changes said shared resources (col. 3 line 54 - col. 4 line 2; col. 4 lines 34-45), said operating system comprising a synchronization algorithm for synchronizing multiple threads of operation with a single thread so as to achieve mutual exclusion between multiple threads performing work on said shared resource (col. 4 lines 37-45; col. 6 lines 10-21) and a single thread changing the state of said shared resource without requiring serialization of all threads (col. 5 lines 34-45; col. 6 lines 10-21) such that an update or change of the state of the shared resource may be made by the single thread only when none of the multiple threads are processing work on the shared resource (col. 6 lines 10-21; col. 7 lines 14-39; col. 11 lines 37-47).

22. Spix teaches the invention as claimed, including the following limitations not shown by Lindholm:

the shared resources being work queues (col. 15 lines 3-27).

23. Tillier teaches the invention as claimed, including the following limitations not shown by Lindholm or Spix:

a network, comprising:

a switched fabric (col. 5 lines 14-36);

remote systems attached to said switched fabric (col. 2 lines 20-40); and

a host system comprising multiple processors; a host-fabric adapter provided to interface with said switched fabric and included work queues each configured to send and receive message data from a single remote system, via said switched fabric (Fig. 1).

24. As per claim 12, Lindholm teaches the invention as claimed, including the network as claimed in claim 11, wherein said synchronization algorithm is executed to synchronize any thread wishing to update or change a state or said work queues with all the threads processing I/O operations on said work queues (col. 6 line 56 - col. 7 line 6).

25. As per claim 17, Tillier teaches the invention as claimed, including the following limitations not shown by the modified Lindholm, specifically a network as claimed in claim 11, further comprising data channels formed between said system and said remote system, via said switched fabric, and supported by the "*Virtual Interface [VI] Architecture Specification*" and the "*Next Generation Input/Output [NGIO] Specification*" for message data transfers between said host system and said remote systems (col. 5 lines 14-36).

26. As per claim 18, Lindholm teaches the invention as claimed, including the network as claimed in claim 11, wherein said synchronization algorithm is installed as

part of a software driver module of an operating system [OS] kernel or an user-level application of said host system (col. 3 lines 31-33).

27. As per claim 19, Tillier teaches the invention as claimed, including the network as claimed in claim 11, wherein said host system and said remote systems represent channel endpoints of a data network implemented in compliance with the "*Next Generation Input/Output [NGIO] Specification*", and data channels formed between said host system and said remote systems, via said switched fabric, are supported by the "*Virtual Interface [VI] Architecture Specification*" and the "*Next Generation Input/Output [NGIO] Specification*" for message data transfers between said host system and said remote systems (col. 5 lines 14-36).

28. **Claims 13-16 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lindholm in view of Spix in view of Tillier in view of Kerrigan.**

29. As per claim 13, Kerrigan teaches the invention as claimed, including the following limitations not shown by the modified Lindholm:

the network as claimed in claim 11, wherein said synchronization algorithm is executed to allow worker threads to work concurrently while processing I/O operations in exclusion of an update thread when the state of said work queues is not changing, and allow an update thread to change the state or update said work queues in exclusion of multiple worker threads (col. 24 lines 28-39).

30. It would have been obvious to one of ordinary skill in the art to combine Lindholm, Spix, and Tillier with Kerrigan since the method taught by the combination of Lindholm, Spix, and Tillier allows all threads that are synchronized with an object to perform state updates on the shared resource. This can lead to excessive context switching and tremendous overhead, a situation that requires remedy. Kerrigan discloses a way of updating an application using an update thread and a synchronization construct, specifically a semaphore. Although Kerrigan is related to updating the data pertaining to an application, the idea is easily combinable with Lindholm since Kerrigan teaches the invention as claimed, including threads and synchronization constructs, as does Lindholm. The combination therein would thus allow Lindholm to be modified such that only one specific thread performs updates on the shared resource, thereby reducing the costly overhead incurred if each thread working on a shared resource updated the status upon synchronization or desynchronization.

31. As per claim 14, Lindholm teaches the invention as claimed, including the network as claimed in claim 11, wherein said synchronization algorithm is executed to support a worker thread operation for processing simultaneous I/O operations on said work queues while concurrently supporting an update thread operation for updating or changing the state of said work queues (col. 1 lines 30-45).

32. As per claim 15, Lindholm teaches the invention as claimed, including a network as claimed in claim 14, wherein said worker thread operation is invoked by one of an event and a user's request, and is performed by:

determining whether a lock is available (col. 5 lines 7-20);

if the lock is not available, waiting until the lock becomes available (col. 6 lines 22-34);

if the lock is available, seizing the lock while incrementing a count by a discrete constant to indicate the number of worker threads that are active, and then releasing the lock after the count has been incremented (col. 11 line 65 - col. 12 line 3);

after the lock has been released, allowing multiple worker threads to process work concurrently (col. 3 line 55 - col. 7 line 20);

determining next whether there is work to be processed (col. 7 lines 23-29);

if there is work to be processed, processing the work until there is no work to be processed (col. 7 lines 23-29); and

if there is no work to be processed, decrementing the count by a discrete constant to indicate when all the worker threads are done with completion processing (col. 7 line 47 - col. 8 line 19).

33. As per claim 16, Lindholm teaches the invention as claimed, including a network as claimed in claim 6, wherein said update thread operation is invoked by a user's request, and is performed by:

determining whether a lock is available (col. 5 lines 7-20);

if the lock is not available, waiting until the lock becomes available when released by any one of the worker threads (col. 6 lines 22-34);

if the lock is available, seizing the lock until the count becomes zero (0) to indicate that it is safe to update or change the state of said shared resource, and updating or changing the state of said shared resource (col. 7 line 63 - col. 8 line 19); and

after said shared resource has been updated, releasing the lock so as to allow either new worker threads to continue I/O operation processing or a different update thread to continue shared resource updating (col. 7 line 63 - col. 8 line 19).

34. **Claims 20-21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lindholm in view of Spix in view of Kerrigan.**

35. As per claim 20, Lindholm teaches the invention as claimed, including a process of synchronizing an update thread which updates a list of shared resources with multiple worker threads which operate on items in the list of shared resources in a multiprocessor system, comprising:

allowing a group of worker threads to concurrently access the list of shared resources to process I/O operations in mutual exclusion, when states of the work queues are not changing (Fig. 3);

incrementing a count of threads processing I/O operations each time a worker thread is running (col. 11 line 65 - col. 12 line 3), while decrementing the count of threads processing I/O operations each time a worker thread is done processing I/O operations (col. 7 line 47 - col. 8 line 19);

when the count of threads reaches a designated value indicating that no worker threads are running, allowing an update to access and update the list of shared resources in exclusion of new worker threads from processing I/O operations (col. 7 line 63 - col. 8 line 19); and

after the list of shared resources is updated, allowing new worker threads to perform I/O operations until all worker threads are done processing I/O operations (col. 7 line 63 - col. 8 line 19).

36. Spix teaches the invention as claimed, including the following limitations not shown by Lindholm:

the shared resources being work queues (col. 15 lines 3-27).

37. Kerrigan teaches the invention as claimed, including the following limitations not shown by Lindholm or Spix:

updating the work queues should be done using an update thread (col. 24 lines 28-39).

38. It would have been obvious to one of ordinary skill in the art to combine Lindholm, Spix, and Kerrigan since the method disclosed in Spix, while disclosing a way of allowing multiple CPUs to access shared resources that include work queues, does so according to an 'anarchistic' scheduling algorithm, which may be unsuitable for most needs. Thus, the combination with Lindholm provides a rigid scheduling algorithm that regulates access to synchronization constructs, while utilizing shared resources of particular I/O devices that support work queues,, such as those associated with network communication. Additionally, Lindholm and Spix allow all threads that are synchronized with an object to perform state updates on the shared resource. This can lead to excessive context switching and tremendous overhead, a situation that requires remedy. Kerrigan teaches a way of updating an application using an update thread and a synchronization

construct, specifically a semaphore. Although Kerrigan is related to updating the data pertaining to an application, the idea is easily combinable with Lindholm since Kerrigan teaches the invention as claimed, including threads and synchronization constructs, as does Lindholm. The combination therein would thus allow Lindholm to be modified such that only one specific thread performs updates on the shared resource, thereby reducing the costly overhead incurred if each thread working on a shared resource updated the status upon synchronization or desynchronization.

39. As per claim 21, Lindholm teaches the invention as claimed, including a computer readable medium that stores computer executable instructions for implementing the process of claim 20 (col. 4 lines 37-45).

Response to Arguments

40. Applicant's arguments filed May 13, 2004 have been fully considered but they are not persuasive.

41. Applicant argues on page 3, "*Lindholm et al. does not disclose synchronizing multiple threads of operation with a single thread to achieve mutual exclusion between the multiple threads performing work on the shared resource and a single thread updating or changing the state of the shared resource as claimed in independent claim 1, for example. The mutex disclosed in Lindholm et al. is locked when it is allocated and has its synchronizers list (or synchronizer identifier) updated to identify the thread that is synchronized with an object and unlocked when it is de-allocated. This is different than*

the present invention as claimed, which allows processors to perform work on a shared resource concurrently while supporting state changes or updates of the shared resources using a synchronization algorithm to synchronize multiple threads of operation with a single thread."

42. Examiner respectfully disagrees with Applicant's characterization of Lindholm. Specifically, the mutex operation discussed in the above argument merely refers to the status of the internal data structure associated with the mutex upon a thread synchronizing itself with the mutex. The operation of the synchronization process is the relevant portion of Lindholm, as in how the synchronization algorithm locks the shared resource. The requirement of the present invention is that a single thread performs state changes and updates the status of the shared resource, while multiple threads may perform work on the shared resource. The synchronization algorithm of Lindholm meets this requirement by providing a cache manager software module that handles the allocation and deallocation of shared resources. When a worker thread seeks to lock a shared resource, the cache manager is invoked to check the status of the resource and perform the necessary steps to allocate the shared resource to the requesting thread in turn. Similarly, when a worker thread is finished with a resource, the cache manager is invoked to make the changes necessary to indicate that the resource is now available. Additionally, the cache manager is implemented as a monitor that continually checks the status of shared resources (col. 1 lines 6-13), and therefore should be implemented as a thread such that the execution of the cache manager is ongoing.

43. Applicant argues on pages 3-4, "*In addition, it would not have been obvious to combine the references as asserted by the Examiner, except in hindsight in view of the present application. Even assuming that one of ordinary skill in the art would have been motivated to combine the references relied upon by the Examiner, at best, one might come up with a system with multiple threads and a shared resource in which only one thread worked on the resource at a time, or a system with one master thread executing an application to generate data for respective slave threads.*"

44. Examiner respectfully disagrees. Motivation to combine the cited references has been thoroughly provided above, using the teachings of the references themselves, and was not arrived at via hindsight in view of the Applicant's disclosure. How the claim limitations are met by the prior art of record is presented thoroughly above, as well as the motivation to combine the references.

Conclusion

45. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Syed J Ali whose telephone number is (703) 305-8106. The examiner can normally be reached on Mon-Fri 8-5:30, 2nd Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai T An can be reached on (703) 305-9678. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Syed Ali
July 30, 2004

MENG-AL T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100